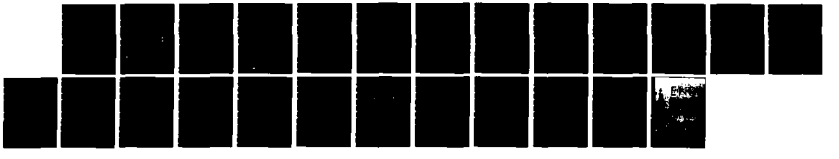
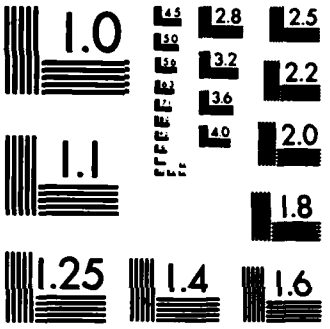
AD-A131 824 ADAPTIVE FINITE ELEMENT METHODS FOR PARABOLIC PARTIAL DIFFERENTIAL EQUATI..(U) RENSSELAER POLYTECHNIC INST TROY NY DEPT OF MATHEMATICAL SCIE.. 1/1

UNCLASSIFIED J E FLAHERTY ET AL. MAY 83 AFOSR-TR-83-0689 F/G 12/1 NL





UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(2)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFOSR-TR- 83-0689	2. GOVT ACCESSION NO.: AD-A131824
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): ADAPTIVE FINITE ELEMENT METHODS FOR PARABOLIC PARTIAL DIFFERENTIAL EQUATIONS	5. TYPE OF REPORT & PERIOD COVERED: Interim
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Joseph E. Flaherty, J. Michael Coyle, Raymond Ludwig, Stephen F. Davis	8. CONTRACT OR GRANT NUMBER(s): AFOSR 80-0192
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Rensselaer Polytechnic Institute, Department of Mathematical Sciences, Troy, N.Y. 12181	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102F, 9749-03
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Office of Scientific Research (NM), Bolling AFB, Washington, D.C. 20332	12. REPORT DATE: May 1983
	13. NUMBER OF PAGES: 21
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

DTIC ELECTE AUG 26 1983 B

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

to appear in the Proceedings of the ARO Workshop on Adaptive Methods for Partial Differential Equations, held 14-16 February 1983, University of Maryland

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Adaptive methods, parabolic partial differential equations, finite element methods

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

We discuss a finite element method for solving initial-boundary value problems for vector systems of partial differential equations in one space dimension and time. The method automatically adjusts the computational mesh as the solution evolves in time so as to approximately minimize the local discretization error. We are thus able to calculate accurate solutions with fewer elements than would be necessary with a uniform mesh.

(continued on back)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (continued)

Our overall method contains two distinct steps: a solution step and a mesh selection step. We solve the partial differential equations using a finite element-Galerkin method on trapezoidal space-time-elements with either piecewise linear or cubic Hermite polynomial approximations. A variety of mesh selection strategies are discussed and analyzed. Results are presented for several computational examples.

"Adaptive Finite Element Methods for Parabolic Partial Differential Equations" by Joseph E. Flaherty*, J. Michael Coyle*, Raymond Ludwig*, and Stephen F. Davis**

The authors

Abstract. ~~We~~ discuss a finite element method for solving initial-boundary value problems for vector systems of partial differential equations in one space dimension and time. The method automatically adjusts the computational mesh as the solution evolves in time so as to approximately minimize the local discretization error. We are thus able to calculate accurate solutions with fewer elements than would be necessary with a uniform mesh.

~~Our~~ This overall method contains two distinct steps: a solution step and a mesh selection step. They solve the partial differential equations using a finite element-Galerkin method on trapezoidal space-time-elements with either piecewise linear or cubic Hermite polynomial approximations. A variety of mesh selection strategies are discussed and analyzed. Results are presented for several computational examples.

1. Introduction. We consider adaptive finite element procedures for finding numerical solutions of M-dimensional vector systems of partial differential equations that have the form

$$(1.1) \quad Lu := u_t + f(x,t,u,u_x) - [D(x,t,u)u_x]_x = 0, \quad a < x < b, \quad t > 0,$$

subject to the initial and linear separated boundary conditions

$$(1.2a) \quad u(x,0) = u_0(x), \quad a \leq x \leq b,$$

* Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, New York 12181. The work of these authors was partially supported by the U. S. Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant Number AFOSR 80-0192 and the U. S. Army Research Office under Grant Number DAAG29-82-K-0197.

** Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, Virginia 23665. The work of this author was partially supported by NASA Contract Number NAS1-17070.

$$(1.2b)\quad B_1u(a,t) := A_{11}(t)u(a,t) + A_{12}(t)u_x(a,t) = b_1(t),$$

$$(1.2c)\quad B_2u(b,t) := A_{21}(t)u(b,t) + A_{22}(t)u_x(b,t) = b_2(t),\ t > 0.$$

There are k_1 initial boundary conditions (1.2a) and k_2 terminal boundary conditions (1.2b). We are primarily concerned with parabolic problems where D is positive definite and $k_1 = k_2 = M$; however, we do not restrict ourselves to this case, but instead we assume that conditions are specified so that equations (1.1) and (1.2) have an isolated solution.

Problems having the above form arise in many applications and our ultimate goal is to create reliable and robust software that will solve a wide class of them without requiring users to supply numerical data such as temporal and spatial step sizes. Thus, we envision a computer code that will automatically discretize and solve (1.1), (1.2) on a nonuniform computational net and attempt to meet a prescribed error tolerance.

The key decisions that must be made in developing such a code are selecting (i) spatial and temporal numerical integration methods, (ii) error indicator and estimator procedures, and (iii) adaptive static and dynamic mesh allocation and distribution techniques.

We discretize and solve (1.1), (1.2) using a finite element-Galerkin method on trapezoidal space-time elements. A detailed discussion and analysis of this approach was given in Davis [6] and Davis and Flaherty [7], and herein we shall only repeat (cf. Section 2) those features that are necessary to the continuity of this paper. Our technique is similar to that of Jamet and Bonnerot [11], and we chose it because it is generally easier to generate high order approximations to partial differential equations on a nonuniform mesh by using finite element methods than by using finite difference methods.

We combine the error indication and static and dynamic mesh adaptation steps by moving a fixed number of finite elements so that they approximately minimize the discretization error per time step. This task is known (cf. de Boor [8] or Pereyra and Sewell [15]) to be asymptotically equivalent to selecting a mesh that equidistributes the error, i.e., a mesh where the error is equal on every element.

Our approach is somewhat similar to the work of Miller et al. [9,13,14], except that they couple the finite element solution and mesh adaptation steps whereas we consider them as distinct phases.

In contrast to these two methods, Bieterman and Babuska [2,3] use a finite element method of lines. In this approach the partial differential equations are first discretized in space using a Galerkin method. This yields a system of ordinary differential equations in time which

may be solved using one of the many available ordinary differential equations codes. They also add or delete elements in regions where the spatial discretization error is estimated to be too large or too small. Their method can potentially solve partial differential equations to a prescribed level of accuracy, whereas this is generally not possible with moving mesh methods that use a fixed number of elements. On the other hand, moving mesh methods can follow evolving nonuniformities and very effectively reduce dispersive errors (cf. Hedstrom and Rodrique [10]). Quite clearly, a code that had both mesh moving capabilities and the ability to add and delete elements would be an ideal software tool for solving partial differential equations. We have been experimenting with adding elements as the temporal integration proceeds in example 3 of Section 4.

In Section 3 of this paper we discuss several mesh equidistribution algorithms and explore their properties. We, unfortunately, show that many algorithms that are based on integrating ordinary differential equations for the mesh velocities are unstable for dissipative partial differential equations. In Section 4, we apply our methods to several examples and in Section 5 we discuss our results and suggest some extensions and improvements.

2. Finite Element Solution Procedure. We discretize problem (1.1)-(1.2) on the strip

$$S_n := \{(x,t) \mid a \leq x \leq b,\ t_n \leq t \leq t_{n+1}\}, \tag{2.1}$$

using a finite element-Galerkin procedure. Hence, we approximate $u(x,t)$ on S_n by $U(x,t) \in \mathcal{U}_K$ and select "test" functions $V(x,t) \in \mathcal{V}_K$, where $\mathcal{U}_K$ and $\mathcal{V}_K$ are K-dimensional spaces of $C^0(S_n)$ functions. We then multiply (1.1) by V, replace u by U, integrate over S_n, and integrate the time derivative and diffusive terms by parts to obtain the following marching problem for determining $U(x,t)$ in successive strips S_n, $n = 0,1,\ldots$:

$$U(x,0) = Pu_0(x),\ a \leq x \leq b,\ n = 0, \tag{2.2a}$$

$$\begin{aligned} F(V,U) := &\int_{t_n}^{t_{n+1}} \int_a^b \{-V^T U_t + V^T f(x,t,U,U_x) + V_x^T D(x,t,U) U_x\}dxdt \\ &+ \int_a^b V^T U dx \Big|_{t_n}^{t_{n+1}} - \int_{t_n}^{t_{n+1}} V^T D(x,t,U) U_x\, dt \Big|_a^b = 0, \end{aligned} \tag{2.2b}$$

$$\forall V \in \mathcal{V}_K,\ (x,t) \in S_n,\ n \geq 0$$

Here, P is an interpolation operator on the space $\mathcal{U}_K$ and U must also satisfy any essential (Dirichlet) boundary condition in (1.2b,c).

In order to select finite element bases for U_K and V_K we partition S_n into N trapezoids T_i^n, $i = 1,\ldots,N$, where T_i^n is the trapezoid with vertices (x_{i-1}^n, t_n), (x_i^n, t_n), (x_{i-1}^{n+1}, t_{n+1}), (x_i^{n+1}, t_{n+1}) (cf. Figure 1).

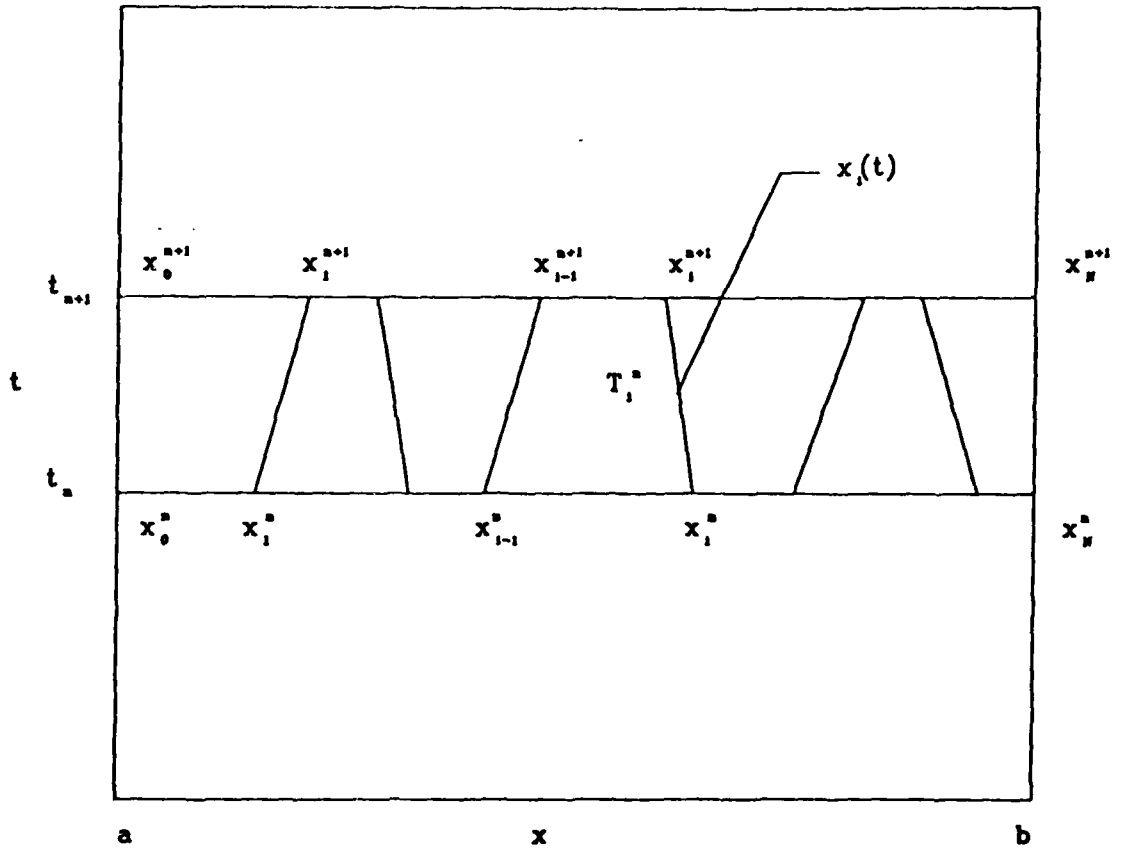


Figure 1. Space-time discretization for time step $t_n \leq t \leq t_{n+1}$.

We write $U \in S_n$ as

$$U(x,t) = \sum_{i=0}^{K} c_i(t)\phi_i(x,t) \qquad (2.3)$$

where each $\phi_i(x,t)$ is selected to be nonzero only on $T_{i-1}^n \cup T_i^n$.

Specifically, we map each T_i^n in the (x,t)-plane into the rectangle

$$(2.4) \qquad R = \{(\xi,\tau) \mid -1 \leq \xi \leq 1,\ 0 \leq \tau \leq 1\}$$

in the (ξ,τ)-plane and, at present, we choose $\phi_j(x,t)$, $j = 0,1,\ldots,K$ to be either piecewise C^0 linear or a piecewise C^1 Hermite cubic polynomials in ξ on T_i^n. We also select $\phi_j(x,t)$, $i = 0,1,\ldots,K$ as a basis for U_K; thus, the dimension of U_K and $_K$ is either N or 2N for linear or cubic approximations, respectively.

The integrals in Eq. (2.2b) are transformed element-by-element into integrals over R and are evaluated numerically. We use the Trapezoidal rule to evaluate the τ integrals and a three-point Gauss-Legendre rule to evaluate the ξ integrals. The resulting system of nonlinear algebraic equations is solved by Newton's method, with users supplying formulas for the Jacobians f_u, f_{u_x}, and D_u. Additional details on our finite element discretization may be found in Davis and Flaherty [7].

3. Adaptive Mesh Selection Strategies. In this section we discuss several algorithms for moving the mesh so that the spatial discretization error in L_2 is approximately minimized at $t = t_{n+1}$. It is known (cf., e.g., [17]) that the spatial error in finite element-Galerkin methods for problems like (1.1.2) satisfies an estimate of the form

$$(3.1) \qquad \|u-U\| \leq C\|u-Pu\| ,$$

where $Pu \in U_K$ interpolates the solution. If we assume that $u(x,t) \in C^k[a,b]$, then Pereyra and Sewell [16] show that the interpolation error in (3.1) can be asymptotically minimized in L_2 for piecewise polynomial interpolants of degree $k-1$ by selecting the interpolation points $x_i(t)$, $i = 0,1,\ldots,N$, at time t such that

$$(3.2) \qquad h_i(t)^k g(\zeta_i,t) = E(t),\ i = 1,2,\ldots,N .$$

Here,

$$(3.3a,b) \quad h_i(t) = x_i(t) - x_{i-1}(t),\ g(x,t) = \{[u^{(k)}(x,t)]^T[u^{(k)}(x,t)]\}^{1/2},$$

$u^{(k)}$ is the kth derivative of u with respect to x, $\zeta_i \in (x_{i-1}, x_i)$, E(t) is an undetermined function of t, and $x_i(t)$ is the line joining x_i^n and

x_i^{n+1} (cf. Figure 1).

The result (3.2) states that the interpolation error is asymptotically minimized when the mesh is moved so as to equidistribute the local spatial discretization error. Many computer codes for two-point boundary value problems use equidistribution algorithms to adapt their computational mesh (cf. Childs et al; [4]). Additional success has been reported in using equidistribution algorithms for variable knot spline interpolation (cf., e.g., de Boor [8]).

In [7], Davis and Flaherty solved (3.2) by an iterative technique. Herein we discuss an alternate procedure which is more restrictive, but has a simpler structure. To begin, we follow de Boor [8] and take the kth root of (3.2) and write it in the asymptotically equivalent form

$$(3.4) \qquad \int_{x_{i-1}}^{x_i} g(x,t)^{1/k}\, dx = c(t),$$

where $c(t)^k \approx E(t)$. We let

$$(3.5) \qquad T(x,t) = \int_a^x g(s,t)^{1/k}\, ds \ .$$

Then

$$(3.6) \qquad c(t) = (1/N)T(b,t)$$

and the equidistributing mesh x_i, $i = 0,1,\ldots,N$, is determined as the solution of the nonlinear system

$$(3.7) \qquad T(x_i,t) = ic(t) \ , \ i = 0,1,\ldots,N \ .$$

Of course, $u^{(k)}$ is unknown and it must be approximated by U. To this end, suppose that we have computed a finite element solution $U(x,t_n)$ at time t_n and on the mesh x_i^n, $i = 0,1,\ldots,N$. We differentiate $U(x,t_n)$ once for piecewise linear approximations or thrice for Hermite cubic approximations and find piecewise constant approximations for $U'(x,t_n)$ or $U'''(x,t_n)$, respectively. We then use finite difference approximations of these derivatives to compute values of $U^{(k)}(x_i,t_n)$ and $g(x_i t_n)$ (cf. (3.3b)), for $i = 0,1,\ldots,N$ and $k = 2$ or 4. We experimented with three, four, and five point difference formulas and found that the five point formulas gave marginally better results, so we are using them in the current version of our code.

We further assume that $g(x,t)^{1/k}$, $k = 2$ or 4, is a piecewise linear function of x with respect to the mesh x_i^n, $i = 0,1,\ldots,N$, and integrate it to find a piecewise parabolic approximation to $T(x,t_n)$ from (3.5).

Finally, we find $c(t_n)$ using (3.6) and determine an approximate equidistributed mesh $\hat{x}_i^n$, $i = 0, 1,\ldots,N$, at time t_n by solving (3.7) using the quadratic formula.

The entire process can potentially be iterated to find a better mesh; however, given all of the approximations made in evaluating $U^{(k)}(x_i^n,t_n)$, we have only tried this at $t = 0$, where the initial function $u_0(x)$ is known to arbitrary precision.

The equidistribution algorithm has a nonunique solution whenever $g(x,t) \equiv 0$; therefore, we may expect numerical difficulties whenever $g(x,t)$ is small on any subinterval. We combat this problem by imposing a lower bound on g, i.e., we replace $g(x,t)$ by

$$\tilde{g}(x,t) := g(x,t) + \eta \,, \tag{3.8}$$

in Eqs. (3.4) and (3.5). Here, η is a small empirically determined quantity that is discussed further in Davis and Flaherty [7]. Among other things, η insures that the solution of (3.7) is a uniform mesh whenever $g(x,t)$ is small everywhere on $[a,b]$.

Our discussion, thus far, has concerned the computation of an equidistributing mesh at time level t_n where a solution $U(x,t_n)$ has already been computed. To obtain an estimate for an optimal mesh at time t_{n+1} prior to computing the solution there, we extrapolate the optimal grids from previous time levels. At the present time, we are using zero order extrapolation, i.e., $x_i^{n+1} = \hat{x}_i^n$, $i = 0,1,\ldots,N$; however, we are experimenting with several different extrapolation strategies, some of which are discussed in Section 3.1.

3.1. Adaptive Strategies for Mesh Velocities. The zero order extrapolation strategy that was just discussed was applied to several examples (cf. Section 4) and, despite its simplicity, it has worked quite well, even on problems with rapidly moving wave fronts. Nevertheless, we can expect that there will be some problems where it will fail to produce an acceptable mesh. Most of our attempts to use higher order extrapolation produced grids that oscillated wildly from time step-to-time step, even when the solution changed quite little. In an attempt to understand and remedy this phenomenon while simultaneously developing a more dynamic adaptive mesh strategy, we differentiated Eq.

(3.4) with respect to time and obtained the following system for the mesh velocities:

$$(3.9)\quad \dot{x}_i\, g(x_i,t) - \dot{x}_{i-1}\, g(x_{i-1},t) + \int_{x_{i-1}}^{x_i} g_t(x,t)dx = \dot{c}\ ,$$

where $\dot{(\ )} := \partial(\ )/\partial t$. This system offers several advantages when it is used in conjunction with our finite element scheme. For example, we can estimate $U_t(x,t_n)$ while assembling the finite element equations. Then, assuming that x_i^n, $i = 0,1,\ldots,N$, is an equidistributing mesh, we can approximate $g_t(x_i,t_n)$, $i = 0,1,\ldots,N$, using the same finite difference scheme that was used to find $g(x_i,t_n)$, $i = 0,1,\ldots,N$. Having done this, $\dot{c}$ is determined as

$$(3.10)\quad \dot{c} = (1/N)\int_a^b g_t(x,t_n)dx\ ,$$

and the mesh velocities $\dot{x}_i^n$, $i = 0,1,\ldots,N$, follow readily from (3.9) by a procedure similar to the one that we used to find x_i^n, $i = 0,1,\ldots,N$. We can then integrate the mesh velocities using an explicit method for ordinary differential equations and obtain an approximation for an equidistributing mesh x_i^{n+1}, $i = 0,1,\ldots,N$.

However, since our experiments with higher order (multi-level) mesh extrapolation produced unstable results and since this type of extrapolation can be viewed as a consistent numerical approximation to the differential system (3.9), we examine the stability of (3.9) before proceeding further. Our analysis is quite general and is not limited to either the specific form of $g(x,t)$ that is given in (3.3b) or to piecewise linear mesh trajectories.

We assume that $x_i(t)$, $i = 0,1,\ldots,N$, is an equidistributing mesh that exactly satisfies (3.4) and (3.9) and introduce a small perturbation $\delta x_i(0)$, $i = 0,1,\ldots,N$, at $t = 0$ that satisfies

$$(3.11)\quad \int_{x_{i-1}+\delta x_{i-1}}^{x_i+\delta x_i} g(x,0)dx = c(0) + \delta c_i(0),\ i = 0,1,\ldots,N\ .$$

Since x_0 and x_N are fixed, the perturbations must also satisfy

$$(3.12)\quad \delta x_0(t) = \delta x_N(t) = 0\ ,\quad \sum_{i=0}^{N} \delta x_i(t) = 0,\quad \sum_{i=0}^{N} \delta c_i(0) = 0\ .$$

We assume that no additional errors are introduced, i.e., $\dot{\delta c}_i(t) = 0$, $i = 0,1,2,\ldots,N$, for $t > 0$. Thus, the perturbed system satisfies

$$(\dot{x}_i + \dot{\delta x}_i)g(x_i + \delta x_i, t) - (\dot{x}_{i-1} + \dot{\delta x}_{i-1})g(x_{i-1} + \delta x_{i-1}, t) + \int_{x_{i-1}+\delta x_{i-1}}^{x_i+\delta x_i} g_t(x,t)dx = \dot{c}, \quad i = 1,2,\ldots,N-1, \quad t > 0. \tag{3.13}$$

We further assume that $|\delta x_i| \ll 1$, $i = 1,2,\ldots,N-1$ and linearize (3.11) and (3.13) to obtain

$$g(x_i(0),0)\delta x_i(0) - g(x_{i-1}(0),0)\delta x_{i-1}(0) = \delta c_i(0), \tag{3.14a}$$

$$\frac{d}{dt}\left[g(x_i,t)\delta x_i(t) - g(x_{i-1}(t),t)\delta x_{i-1}(t)\right] = 0, \quad i = 1,2,\ldots,N-1. \tag{3.14b}$$

This system is easily integrated to yield

$$\delta x(t) = L^{-1}(t)L(0)\delta x(0), \tag{3.15a}$$

where

$$\delta x(t) = [\delta x_1(t), \delta x_2(t),\ldots,\delta x_{N-1}(t)]^T, \tag{3.15b}$$

$$L(t) = \begin{bmatrix} g(x_1(t),t) & & & \\ -g(x_1(t),t) & g(x_2(t),t) & & \\ & \cdot & \cdot & \cdot \\ & & -g(x_{N-2}(t),t) & g(x_{N-1}(t),t) \end{bmatrix}. \tag{3.15c}$$

Of course, since $L(t)$ is lower triangular, the solution of (3.14) can be written in a more explicit form as

$$\delta x_i(t) = \frac{g(x_{i-1}(t),t)}{g(x_i(t),t)}\,\delta x_{i-1}(t) + \frac{g(x_i(0),0)}{g(x_i(t),t)}\,\delta x_i(0) + \frac{g(x_{i-1}(0),0)}{g(x_i(t),t)}\,\delta x_{i-1}(0), \quad i = 1,2,\ldots,N-1. \tag{3.16}$$

The system (3.9) is stable to linear perturbations when $\delta x(t)$ decays and this occurs when $\|L^{-1}(t)L(0)\| < 1$, for some matrix norm. Unfortunately, the choice of $g(x,t)$ given by Eq. (3.3b), and other reasonable choices, are likely to be decreasing functions of time for dissipative parabolic partial differential equations and this will almost certainly yield a value for $\|L^{-1}(t)L(0)\|$ that is larger than unity.

Local instabilities can also occur when the mesh is moved so that one of the three ratios involving g in Eq. (3.16) exceeds unity. However, since these instabilities are local, they may either grow or decay as time progresses.

The following two examples illustrate some of the instabilities that can occur in Eq. (3.9).

Example 3.1. Consider the constant coefficient heat conduction problem

$$u_t = u_{xx} \ , \ 0 < x < 1 \ , \ t > 0 \ , \tag{3.17a}$$

$$u(x,0) = \sin\pi x \ , \ u(0,t) = u(1,t) = 0, \tag{3.17b,c}$$

which has the exact solution

$$u(x,t) = e^{-\pi^2 t} \sin\pi x \ . \tag{3.17d}$$

Since the solution of this problem is separable, the optimal strategy is to generate an equidistributed mesh at time t = 0 and use it for all time. When $g(x,t)$ is given by (3.3b) we find that $\|L^{-1}(t)L(0)\| \sim e^{-\pi^2 t}$; thus, we expect the solution of (3.9) to be unstable. In Figure 2, we display the meshes produced by both (3.4) and (3.9) for $g(x,t) = |u_{xx}(x,t)|$, and the instability produced by using (3.9) is clearly visible. We note that exact values of u_{xx} were used in obtaining Figure 2 and the only errors that were introduced in the computation were due to trapezoidal rule integration and a perturbation of 0.01 in the initial mesh for Eq. (3.9).

Example 3.2. We consider a constant coefficient heat conduction problem that was studied in Davis and Flaherty [7], i.e.,

$$u_t = u_{xx} + f(x,t) \ , \ 0 < x < 1 \ , \ t > 0 \ , \tag{3.18}$$

The initial conditions, Dirichlet boundary conditions, and source f are chosen so that the exact solution is

$$u(x,t) = \tanh(r_1(x-1) + r_2 t) \ . \tag{3.19}$$

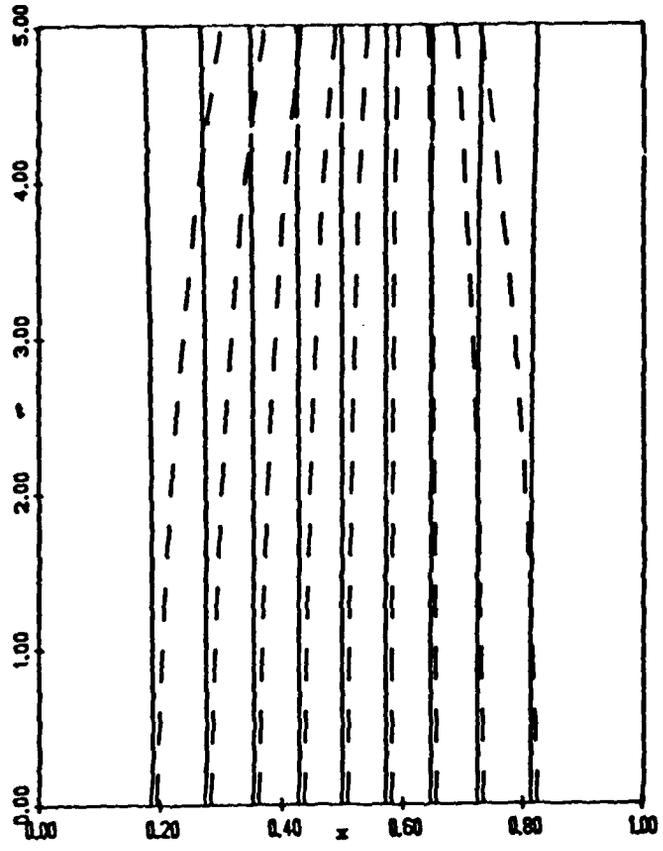


Figure 2. Meshes that were produced by solving Eq. (3.4) (solid line) and integrating Eq. (3.9) (dashed line) for Example 3.1.

The solution (3.19) is a wave that travels in the negative x direction when r_1 and r_2 are positive. The values of r_1 and r_2 determine the steepness of the wave and its propagation speed. The meshes produced by both (3.4) and (3.9) for $r_1 = r_2 = 5$ and $g(x,t) = |u_{xx}(x,t)|$ are shown in Figure 3. The solution of Eq. (3.9) is initially stable, but as time progresses and $g(x,t)$ decays it becomes unstable.

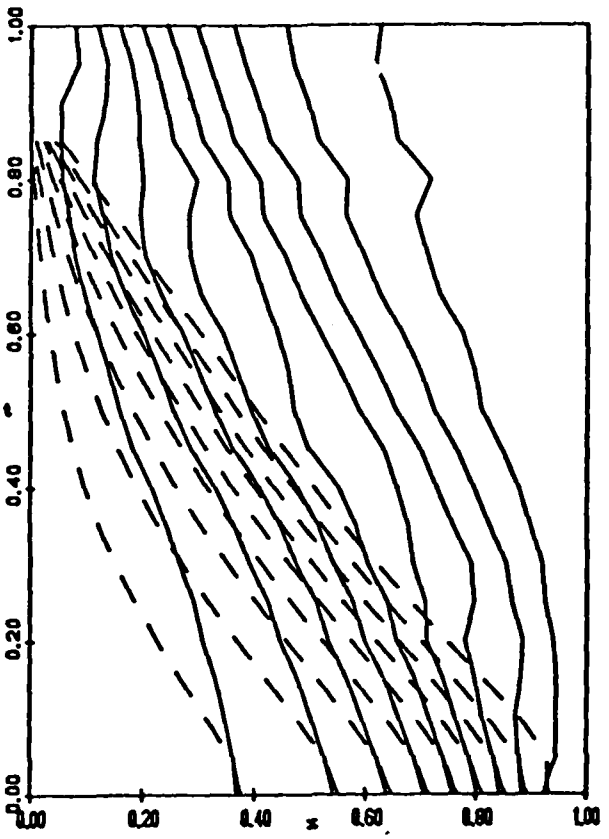


Figure 3. Meshes that were produced by solving Eq. (3.4) (solid line) and integrating Eq. (3.9) (dashed line) for Example 3.2.

Petzold [16] suggested that the following linear combination of Eqs. (3.4) and (3.9) might yield stable meshes with some improved dynamic behavior:

$$\dot{x}_i g(x_i,t) - \dot{x}_{i-1} g(x_{i-1},t) + \int_{x_{i-1}}^{x_i} g_t(x,t)dx + \lambda \int_{x_{i-1}}^{x_i} g(x,t)dx = \dot{c} + \lambda c \ . \tag{3.20}$$

Here $\lambda > 0$ is a parameter to be determined so that (3.20) is stable. We have not done enough analysis or experimentation to form any firm conclusions ; however, in Figure 4 we compare the solution of Eq. (3.4) with Eq. (3.20) for Example 3.2. Equation (3.20) was solved by the explicit Euler method using $\lambda = 1/\Delta t$, uniform time steps of $\Delta t = 1/20$, exact values of g and g_t, and trapezoidal rule integration. It appears that the mesh produced by Eq. (3.20) has some local instabilities, but is globally stable for this example. Hence, this method has promise, but much more testing is needed to determine its behavior, especially when the approximate solution $U(x,t)$ is used to calculate g and g_t.

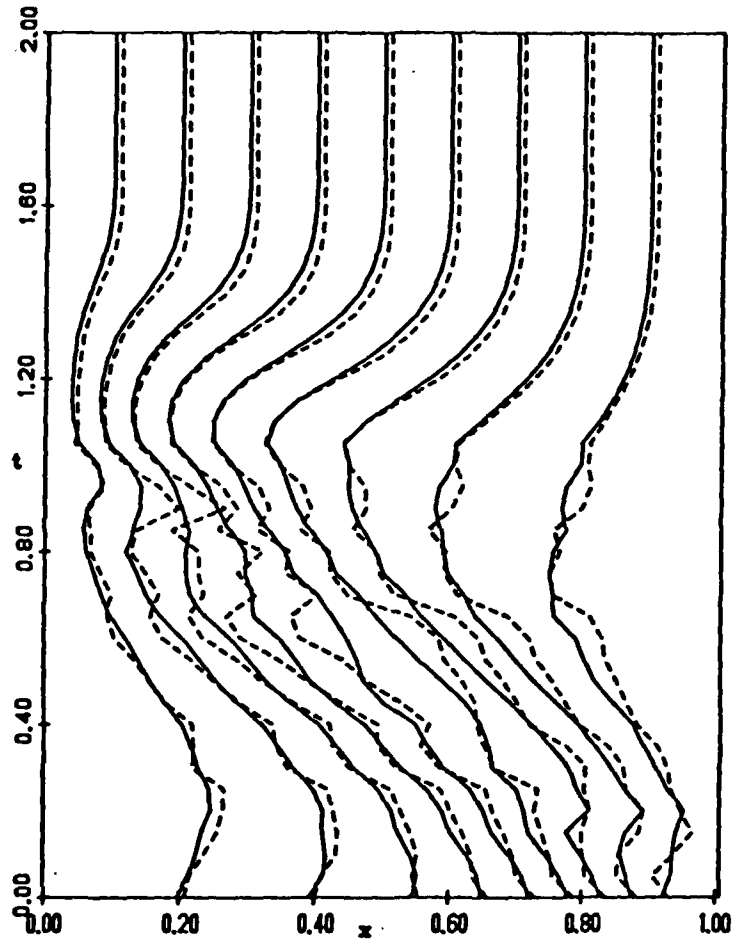


Figure 4. Meshes that were produced by solving Eq. (3.4) (solid line) and integrating Eq. (3.20) (dashed line) for Example 3.2.

4. Examples. In this section we examine the performance of our adaptive finite element method on four examples. In all of these we used the equidistribution algorithm as discussed in Section 3 with zero order extrapolation.

Example 4.1. We consider Example 3.2 (cf. Eqs. (3.18) and (3.19)) with $r_1 = r_2 = 100$. We compare the pointwise errors vs. x at t = 1 for computations performed with uniform and adaptive meshes of 20 finite elements using piecewise linear (cf. Figure 5a) and cubic approximations (cf. Figure 5b). The results were obtained using uniform time steps of $\Delta t = 0.01$ and 0.0025, respectively, for the linear and cubic approximations. We see that the computations using the fixed uniform grids have large errors at the wave front, which is near x = 0 at t = 1. The adaptive mesh algorithm, on the other hand, concentrates elements in the wave front (cf. Figure 3) and distributes the local error more evenly over the domain.

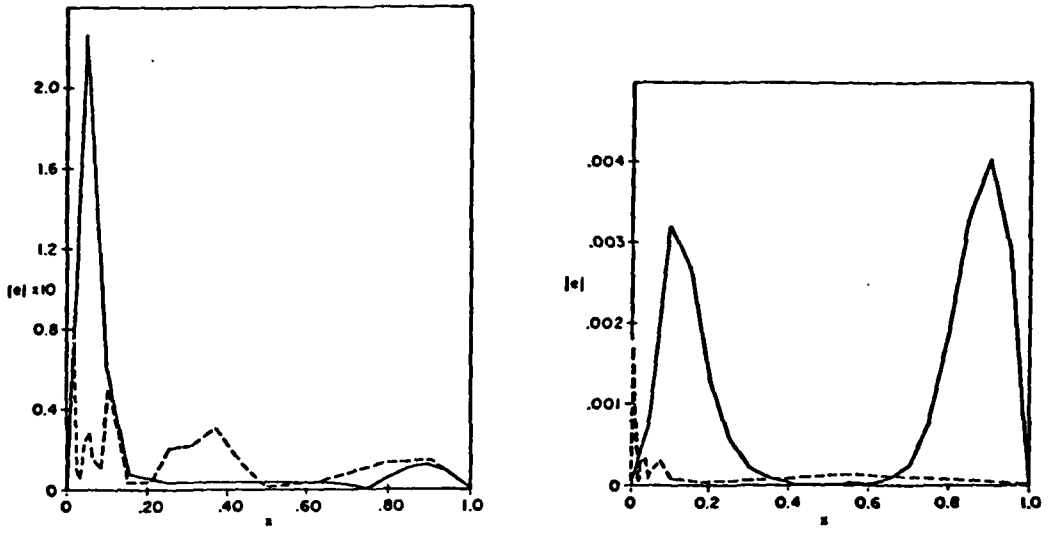


Figure 5. Local error at t = 1.0 for Example 4.1 with 20 elements. The solid curve was computed on a fixed uniform mesh and the dashed curve on an adaptive mesh. The solution in Figure 5a (left) used piecewise linear approximations with uniform time steps of $\Delta t = 0.01$ and that in Figure 5b (right) used piecewise cubic approximations with uniform time steps of $\Delta t = 0.0025$.

Example 4.2. We consider the following problem for Burgers' equation:

$$u_t + uu_x = \varepsilon u_{xx} \ , \ 0 < x < 1 \ , \ t > 0$$
$$u(x,0) = \sin \pi x \ , \ u(0,t) = u(1,t) = 0 \ , \tag{4.1}$$

and $\varepsilon = 5 \times 10^{-3}$. The solution of this problem is a pulse that steepens as it travels to the right until it forms a shock layer at x = 1. After a time of $O(1/\varepsilon)$ the pulse dissipates and the solution decays to

zero. We compare solutions using piecewise Hermite cubic approximations on a uniform and an adaptive mesh of 10 finite elements in Figures 6a and 6b, respectively. Both calculations were performed with uniform time steps of $\Delta t = 0.1$. It is well known that finite difference and piecewise linear finite element solutions of this problem exhibit spurious mesh oscillations unless the mesh width is $O(\epsilon)$ within the shock layer. However, the solution using piecewise cubic approximations on a uniform mesh, that is shown in Figure 6a at $t = 0.6$, is pointwise very accurate and the large errors appear in the slope of the computed solution. These errors largely disappear when the mesh adapts with the solution and is appropriately concentrated in the shock layer (cf. Figure 6b).

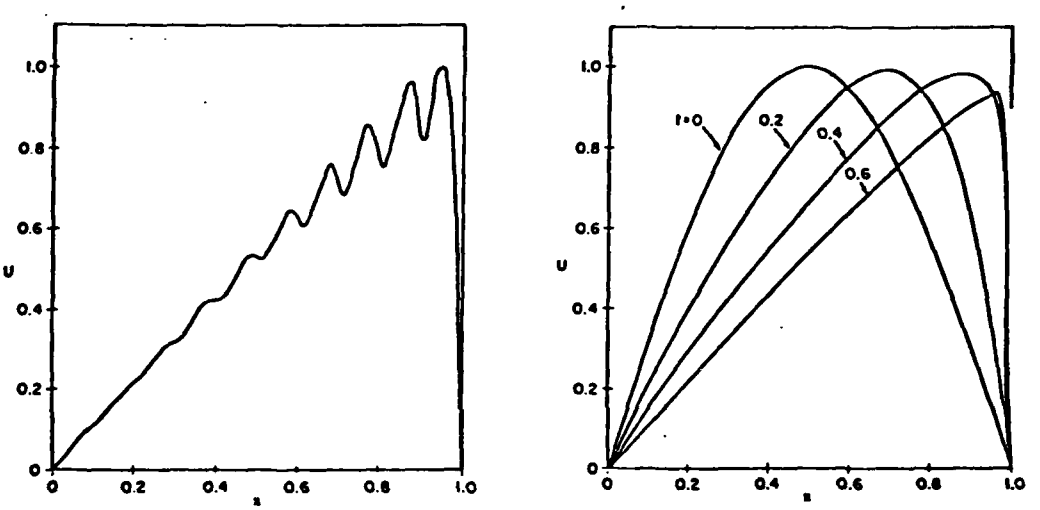


Figure 6. Solution of Example 4.2 using cubic approximations with 10 elements and uniform time steps of $\Delta t = 0.01$. The solution in Figure 6a (left) was obtained using a fixed uniform mesh while that in Figure 6b (right) used an adaptive mesh.

Example 4.3. We are currently investigating the following focusing problem for the nonlinear Schrödinger equation in cylindrical coordinates:

$$(4.2) \quad u_t = (i/r)(ru_r)_r + i|u|^2u \ , \ r > 0 \ , \ t > 0 \ , \ u(r,0) = Ae^{-ar^2} \ ,$$

where $i = \sqrt{-1}$ and $u(r,t)$ is a complex valued function.

It is known (cf. [12]) that the solution of (4.2) "self-focuses", i.e., it develops infinite values of u in a finite time, when the initial conditions are "strong enough". Problems of this type occur in laser optics, and we are trying to determine the conditions which cause the solution to "blow up" and also the local character of the solution just prior to blow up. This problem is still under investigation, in collaboration with A. Newell [5], and herein we only present some preliminary results which, we feel, illustrate the need for adaptive mesh strategies on difficult nonlinear partial differential equations.

In Figures 7a and 7b we illustrate $|u(r,t)|$ for two sets of initial conditions having a = 25 and A = 14 and A = 15, respectively. The solution in Figure 7a does not focus while (we speculate that) the solution in Figure 7b focuses. The dramatic growth in the magnitude of the solution that is shown in Figure 7b is also accompanied by rapid changes in phase as focusing nears.

Example 4.4. We are studying elastic-plastic impact problems for cylindrical rods using the long rod model of T. W. Wright [18]:

$$(4.3) \qquad v = w_t \ , \ e = w_x \ , \ p = u_t \ , \ q = u_x \ ,$$

$$(4.4) \qquad e_t = v_x \ , \ q_t = p_x \ ,$$

$$(4.5) \qquad v_t = S_x \ , \ p_t = 2(Q_x - P) \ .$$

Here, u and w are dimensionless radial and axial displacement components, p and v are radial and axial velocity components, e is the axial strain, q is the shear strain, and S, P, and Q are axial, radial, and shear forces, respectively. Equations (4.3) define the strain and velocity variables in terms of the displacements, Eqs. (4.4) are compatibility relations, and Eqs. (4.5) are the equations of motion. We also need appropriate constitutive laws that relate S, P, and Q to e, u, and q, and herein we simply use the linear Hooke's laws

$$(4.6) \qquad S = e + \frac{2\nu}{1-\nu} u \ , \ P = \frac{2}{1-\nu} (\nu e + u) \ , \ Q = \frac{1-2\nu}{4(1-\nu)} q \ ,$$

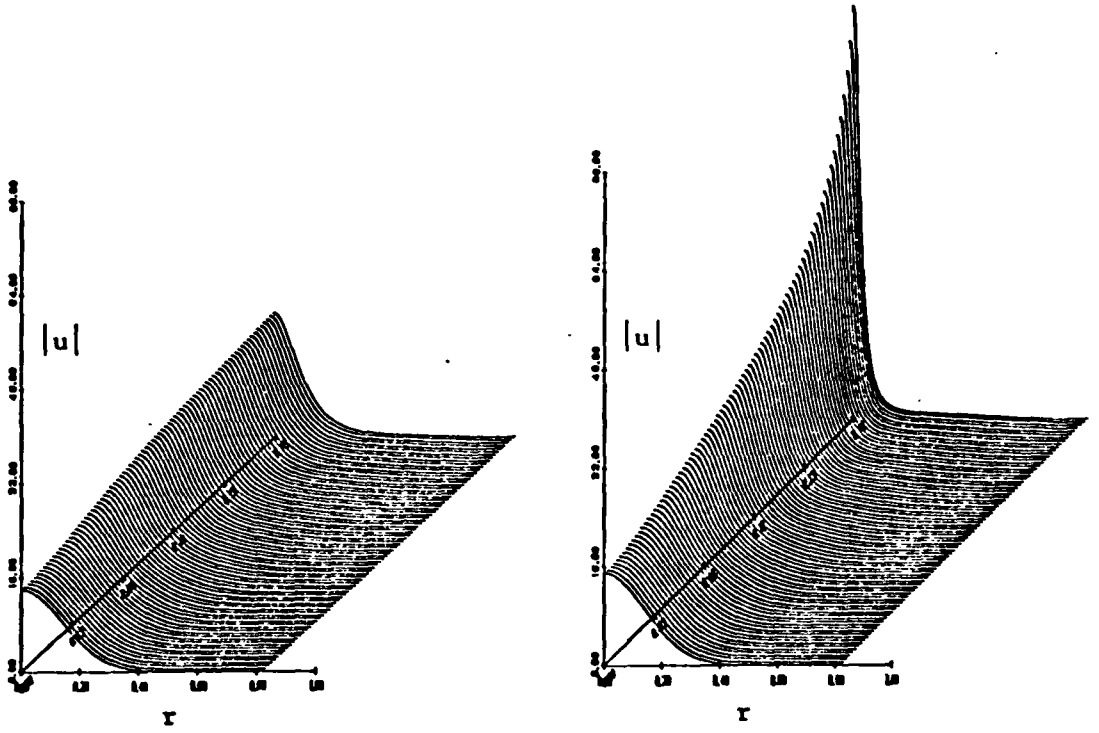


Figure 7. Magnitude $|u|$ of the solution of the Schrödinger equation (4.2) with a = 25 and A = 14 (Figure 7a, left) and A = 15 (Figure 7b, right).

where ν is Poisson's ratio. If $\nu = 0$ we have a one-dimensional theory and Eqs. (4.3)-(4.6) reduce to a simple wave equation. However, if $\nu \neq 0$ Eqs. (4.3)-(4.6) give a two-dimensional theory that is valid when the length L of the cylindrical rod is large compared to unity.

Like the previous example, this is work in progress in collaboration with J. J. Wu and T. W. Wright, so we will only report some preliminary results that illustrate the differences between the one- and two-dimensional theories. We use homogeneous initial conditions and the boundary conditions

(4.7) $v(0,t) = 0.1$, $Q(0,t) = 0$, $S(5,t) = 0$, $Q(5,t) = 0$.

These conditions correspond to a cylinder of length L = 5 that is hit at t = 0 by a rigid wall that is moving with a velocity of ten percent of the longitudinal wave speed of the bar. Since the initial conditions are homogeneous, our adaptive algorithms have no choice but to select a uniform initial mesh. However, the velocities and strains jump at x = 0, t = 0 and the mesh should be concentrated in the vicinity of this point. There are several possibilities for overcoming this difficulty. For example, we could either input a graded initial mesh or we could use nonhomogeneous initial conditions and start the problem at a small value of t > 0. We tried both alternatives, and they gave very similar results for t > 0. We also added artificial viscosity terms εe_{xx}, εq_{xx}, εv_{xx}, and εp_{xx}, respectively, to the right sides of Eqs. (4.4) and (4.5).

We present results for the axial stress S vs. x for t = 0,1,2,...,10 in Figures 8 and 9 for $\nu = 0$ and $\nu = 0.3$, respectively. In each case 50 piecewise linear elements were used with an artificial viscosity coefficient $\varepsilon = 0.005$.

It is known, that one-dimensional elastic waves are non-dispersive whereas two-dimensional waves are dispersive. The dispersive nature of the two-dimensional solution is clearly visible in Figure 9. Unfortunately, the one-dimensional solution is much more dispersive than it should be. We conjecture that the excessive dispersion is due to our form of the artificial viscosity, and we are experimenting with different models.

5. Discussion. The computations of Section 4 show that it is possible to construct an accurate and stable adaptive grid finite element procedure for nonlinear systems of partial differential equations that offers several advantages over fixed grid methods. However, we have much more to do in order to achieve our goal of developing reliable and robust general purpose software for partial differential equations.

In this paper we concentrated on improving the mesh moving capabilities of our code. We have always found equidistribution with zero order extrapolation to be unsatisfying; however, our attempts to use higher order extrapolation always ended in failure. The theoretical results of Section 3 explain why this is so, and indicate a possible remedy that we hope to explore further.

We feel that we have reached the limit in terms of what can be achieved with a fixed number of elements per time step. There are basically two possible ways of extending our methods to include the ability of adding and deleting elements as the integration progresses in time. We can envision a method of lines approach, in the spirit of Bieterman and Babuska [2,3]; however, with "lines" that adaptively equidistribute the local spatial component of the error. Elements can be added or deleted as the integration progresses and the power of the ordinary differential equations codes can still be used.

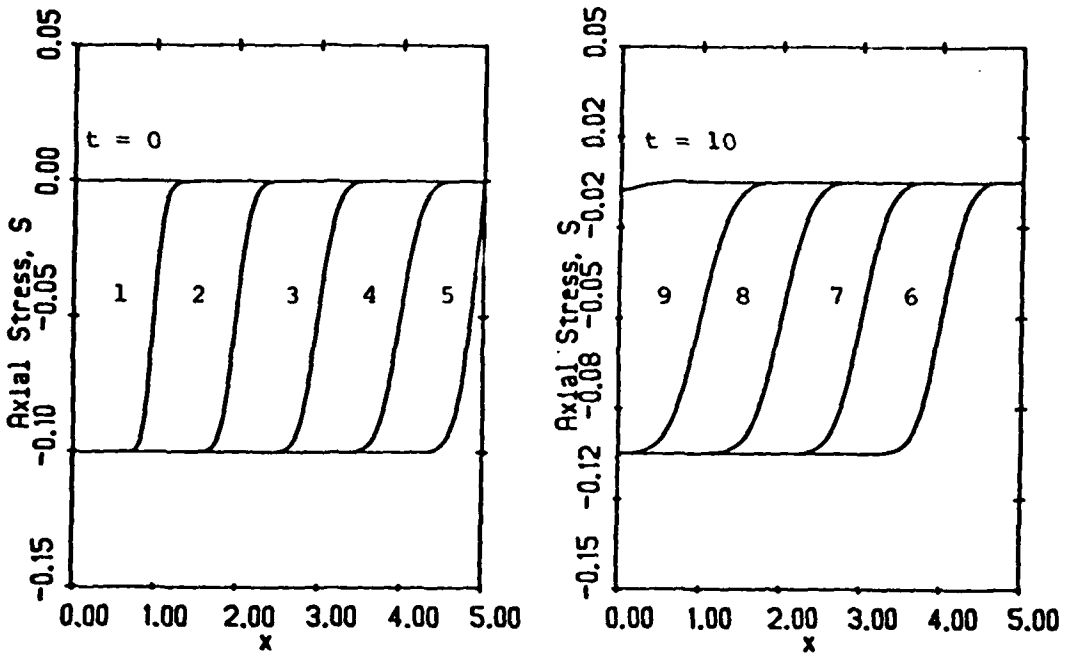


Figure 8. Axial stress S vs. x for t = 0,1,...,10 for Example 4.4 with $\nu = 0$.

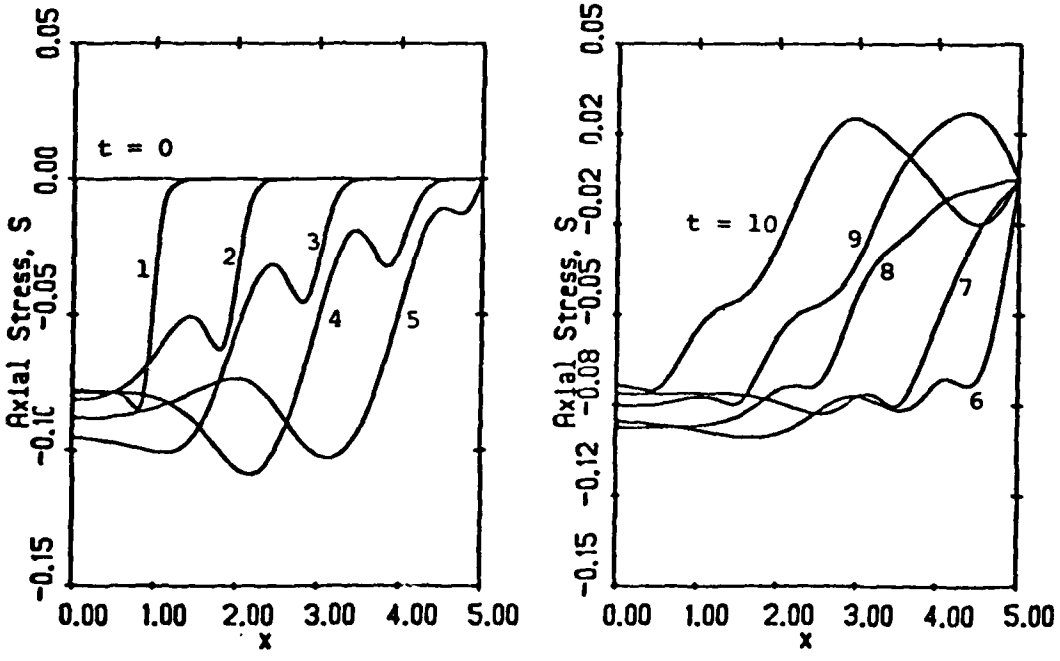


Figure 9. Axial stress S vs. x for t = 0,1,...,10 for Example 4.4 with $\nu = 0.3$.

A second possibility is to locally refine, using space-time-trapezoidal elements in regions having large local error. This is in the spirit of the adaptive finite difference methods of Berger [1]; however, the computational cells would be trapezoidal rather than rectangular.

We are exploring both approaches, but feel that the latter offers the most promise.

Acknowledgment. The authors would like to thank Dr. R. C. Y. Chin of Lawrence Livermore Laboratories and Dr. L. Petzold of Sandia National Laboratories for their helpful comments and suggestions on adaptive mesh algorithms.

REFERENCES

[1] M. J. BERGER, Adaptive mesh refinement for hyperbolic partial differential equations, Report No. STAN-CS-82-924, Department of Computer Science, Stanford University, 1982.

[2] M. BIETERMAN and I. BABUSKA, The finite element method for parabolic equations, I. a posteriori error estimation, Institute for Physical Science and Technology, Laboratory for Numerical Analysis, Tech. Note BN-983, University of Maryland, 1982.

[3] M. BIETERMAN and I. BABUSKA, The finite element method for parabolic equations, II. a posteriori error estimation and adaptive approach, Institute for Physical Science and Technology, Tech. Note BN-984, University of Maryland, 1982.

[4] B. CHILDS, M. SCOTT, J. W. DANIEL, E. DENMAN, and P. NELSON, Codes for Boundary-Value Problems in Ordinary Differential Equations: Proceedings of a Working Conference, May 14-17, 1978, Lecture Notes in Computer Science, No. 76, Springer-Verlag, Berlin, 1979.

[5] J. M. COYLE, J. E. FLAHERTY, and A. C. NEWELL, Self focusing problems for the nonlinear Schrodinger equation, in preparation.

[6] S. F. DAVIS, An adaptive grid finite element method for initial-boundary value problems, Ph.D. Dissertation, Rensselaer Polytechnic Institute, 1980.

[7] S. F. DAVIS and J. E. FLAHERTY, An adaptive finite element method for initial-boundary value problems for partial differential equations, SIAM J. Sci. Stat. Comput. 3 (1982), pp. 6-27.

[8] C. DE BOOR, A Practical Guide to Splines, Applied Mathematical Sciences, No. 27, Springer-Verlag, New York, 1978.

[9] R. J. GELINAS, S. K. DOSS, and K. MILLER, The moving finite element method: Applications to general partial differential equations with multiple large gradients, J. Comp. Phys. 40 (1981), pp. 202-249.

[10] G. W. HEDSTROM and G. H. RODRIQUE, Adaptive-grid methods for time-dependent partial differential equations, UCRL-87242 preprint, Lawrence Livermore National Laboratory, Livermore, 1982.

[11] P. JAMET and R. BONNEROT, Numerical solution of the Eulerian equations of compressible flow by a finite element method which follows the free boundary and the interfaces, J. Comp. Phys. 18 (1975), pp. 21-45.

[12] K. KONNO and H. SUZUKI, Self focusing of laser beam in nonlinear media, Physica Scripta, 20 (1979), pp. 382-386.

[13] K. MILLER and R. N. MILLER, Moving finite elements. I, SIAM J. Numer. Anal. 18 (1981), pp. 1019-1032.

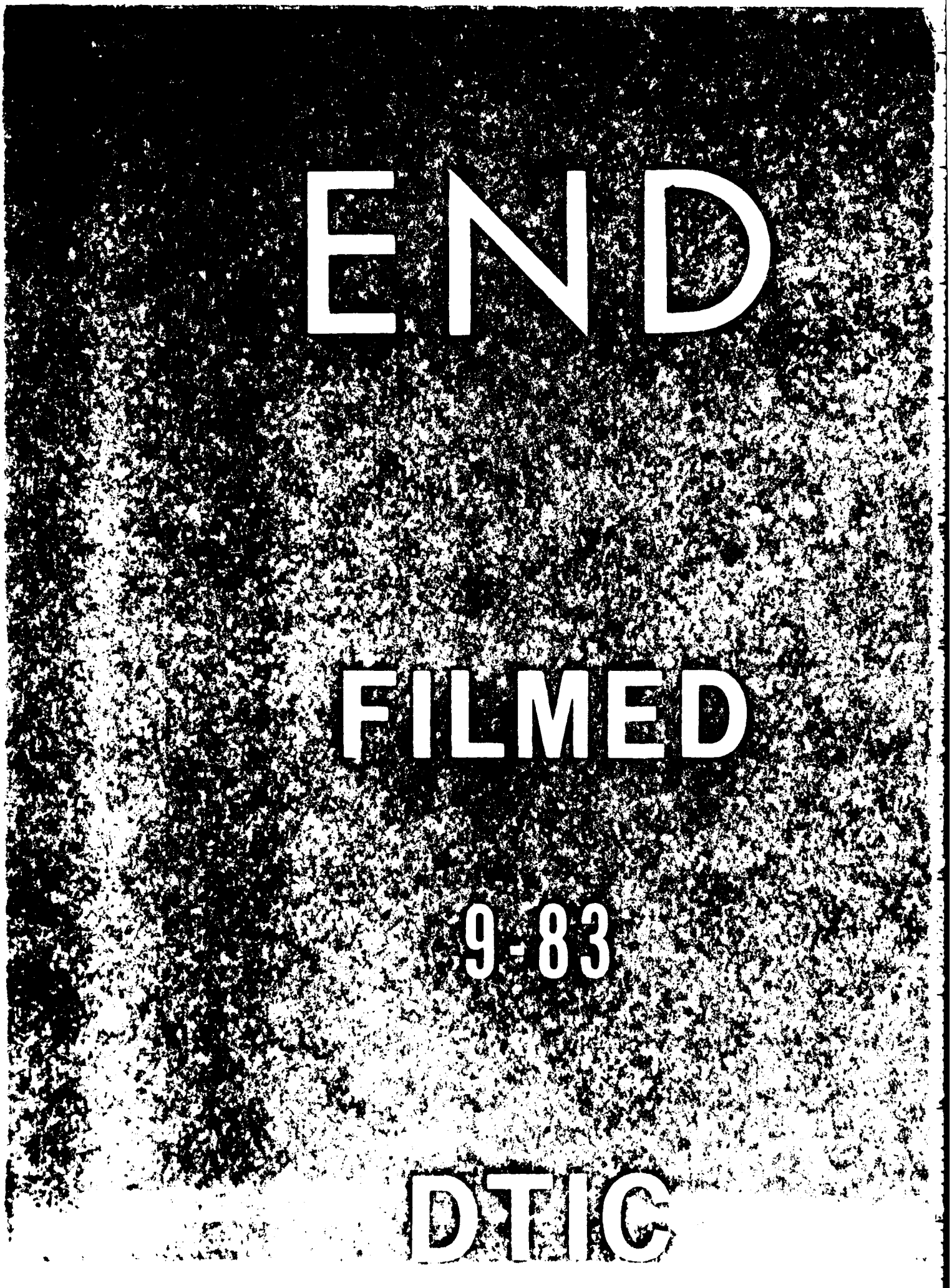
[14] K. MILLER, Moving finite elements. II, SIAM J. Numer. Anal. 18 (1981), pp. 1033-1057.

[15] V. PEREYRA and E. G. SEWELL, Mesh selection for discrete solution of boundary problems in ordinary differential equations, Numer. Math. 23 (1975), pp. 261-268.

[16] L. PETZOLD, Personal communication, 1983.

[17] M. F. WHEELER, A priori L-error estimates for Galerkin approximations to parabolic partial differential equations, SIAM J. Numer. Anal. 10 (1973), pp. 723-759.

[18] T. W. WRIGHT, Nonlinear waves in rods, Tech. Rep. ARBRL-TR-02324, U. S. Army Armament Research and Development Command, Ballistic Research Laboratories, Aberdeen Proving Ground, 1981.


END
FILMED
9-83
DTIC